UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/733,789	12/10/2003	Mehmet Yunt	MWS-087RCE	8948

74321 7590 04/07/2008
LAHIVE & COCKFIELD, LLP/THE MATHWORKS
One Post Office Square
Boston, MA 02109-2127

EXAMINER	
VU, TUAN A	
ART UNIT	**PAPER NUMBER**
2193	
MAIL DATE	**DELIVERY MODE**
04/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/733,789	**Applicant(s)** YUNT ET AL.	
	Examiner Tuan A. Vu	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 January 2008*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-83* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-83* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 1/9/08.

As indicated in Applicant's response, claims 1-83 have been amended. Claims 1-83 are pending in the office action.

Claim Objections

2. Claims 1, 39, 77 are objected to because of the following informalities: the language for describing 'dynamically updated execution list' in terms of (i) depicting an *execution order* of a plurality of methods called during execution of a time step of the model' and (ii) 'to list methods that have been called during the time step until a specified point' is deemed inaccurate syntactic/semantic construct and not easily interpretable. The language used to specify the 'dynamically … execution list' conveys that this list is expressed as both (i) and (ii) during some specific time frame of executing a time step. According to Fig. 12A-B of the Specifications regarding a time step, 'execution order' amounts to time-line ordering of blocks referred to as sample time blocks or method blocks, which represent, for example, output methods for their respective blocks, so that state of change occurs when these *block methods* are executed within a GUI interface for time-line setting. Clearly, 'execution order' of those block methods are time-evolving method blocks representing underlying execution of method whose calls configuration is interacted with by the developer. Further, the phrase 'dynamically updated execution list' or *execution order* cannot be exactly a 'update' or 'output method execution list' as described as algorithmic approach in Fig. 10, 13 of the disclosure, since the phrase does not include either *output* or *update execution list* (see Specifications pg. 34); nor can this very 'dynamically updated execution list' -- or 'execution order' connotation thereof -- be simultaneously equated

to a directory tree or a view of colored nodes thereof (see Fig. 17 and related text) the description of which (emphasis added) does not mention about 'time step' execution as well as 'execution order'. When the claim does not provide an alternative for either (i) or (ii) there would be a large difference between *list 266*, Fig. 15 and execution of blocks order (Specifications: Figs. 7-9, 11-12); making it hard to accept that the both (i) and (ii) can co-exist with a same structural and functional measure for implementing this one 'dynamically … execution list'. That is, depiction of time blocks in time line (Figs. 7-9, 11-12) cannot be analogous to a process algorithm operating on *update/output method execution list* as described in respectively (Fig. 10, 13). Nor can 'dynamically … execution list' be exactly a directory of color nodes called 'execution list view' in Fig. 17, when this directory tree view is deemed very different from the algorithm of Fig. 13-14, and all the more so (emphasis added) from the time-based block execution of methods in Fig. 12 representing what is termed as 'execution order' in the claim. As for *list 266*, Fig. 15, the Specifications describes 'hierarchy view' being updated by means of the user clicking, or whenever a new record therefor is created (Specs, pg. 42, bottom to pg. 43), thus this listing of colored nodes being changed is not clearly 'execution order' in direct association with method invocation <u>during a model time-step execution</u> (as in ii) but rather depends on a record update or user's intervention. In short, *execution order* of blocks under developer's control (Figs. 7-9, 11-12) cannot clearly be equated to directory of colored nodes that vary with simulation, hence (i) and (ii) cannot be represented by one common language but rather have to be described by more distinct terminologies. While the claim (or even the Specifications) does not explicitly prove that 'execution order' (of block methods) and visually listing of methods (that have been called) is a same concept, the language using 'dynamically updated execution list' to encompass both (i) and (ii) without any further details as to possibility of an alternative, is deemed improper choice of language or far-fetched use of terminology, especially when such language choice has no realistic corroboration from the Disclosure. As the language of the claim fails to secure either of the two aspects (i) and (ii) of said 'dynamically updated execution list', broad interpretation is permissible and in light of reasonable understanding of the diverse sections of Disclosure, the above 'execution order' will be treated as **methods represented by visual blocks**; and **listing of** methods called (during a step execution process) will be treated as **depiction of** underlying operations or **invocation of code in terms of method invocations while executing model and/or calls by the tool or calls being reconfigured/modified by the user so as to dynamically show the time-line state or changes of those method blocks**. Appropriate correction is required, lest this type of language impropriety would lead to a § 112, first or second paragraph type of rejection.

3. Claim 8 is objected for a typographical type of error in 'visual indication is made by at least *one of one of altering* the color'

4. claims 2,3,38 are objected to because some syntactic improprieties in '-between' (cl. 2-3) and "one of a set ... or a set of profiling ... are displayed ..." (cl. 38)

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-83 are rejected under 35 U.S.C. 103(a) as being unpatentable over *the Mathworks*, 'Simulink: Model-based and System-based Design', *Using Simulink*, Version 5, copyright 1990-2002, last printed July 2002, ch. 2-11, 13-14; url: <http://aer.ual.es/docencia_es/iai/archivos/simulink.pdf> (hereinafter Simulink5), further in view of Chandhoke et al, USPubN: 2003/0144751(hereinafter Chandhoke).

As per claim 1, Simulink5 discloses a method in a graphical modeling and execution environment, the method comprising:

providing a model view and an execution list view of a model being executed (ch. 2: pg. 2-10→2-24; ch.13: pg. 13-17→13-19), the model view graphically depicting a plurality of components of the model, the execution list view depicting execution list depicting the execution order of methods called (e.g. *Time Step, Math Function block, Sum block, Product block* - pg. 2-10,11; 2-19,20; ch. 5: pg. 5-16→5-17) during the execution of a time step (e.g. ch. 10: pg. 10-40) of the model, the execution list to list methods that have been called during the time step until a specified point in execution of the time step (e.g. ch. 10: pg. 10-40; ch. 5: pg. 5-16→5-24; ch. 13: pg. 13-20-13-26 – Note: depicting of time-based blocks – see pg. 5-17 -each representing a computation reads on list of methods that have been called; block execution order – see pg. 13-21, 13-23 – by user-driven commands or non-virtual execution event reads on list of underlying methods that reflect the dynamic selection by the user calling or directives – refer to Claim Objection); the model view interfaced with a debugger; and

indicating visually a execution list to list the methods that have been called (e.g. ch. 10: pg. 10-40; ch. 5: pg. 5-16→5-24; ch. 13: pg. 13-20-13-26) on the model view at the specified point in the time step (Note: depicting of blocks in a visual debugging tool inherently teaches instant view at a given or specified point in a time step).

However, Simulink5 does not explicitly teach that the execution list depicted during the execution of a time step is **dynamically updated and changing** during the execution of the model, nor does Simulink5 disclose indicating a **state of said dynamically updated** execution list at the specified point. Using a Simulator tool to observe dynamic changes to a graphical representation of the methods being called to provide operational and visual effects was a well-known concept with Simulation methodologies. And Simulink5 provides this dynamic representation of blocks representing the underlying methods being called by the user-input directed simulating environment (see ch. 2-19 to 2.33 – Note: any user readjusting is being reflected in subsequent display of timing diagram of blocks in execution). In an analogous approach as to using Simulator tool to effect simulation of parts of devices as in Simulink5, Chandhoke teaches representing evolution of blocks or operations being simulated according to a stepping paradigm and further provide dynamic redisplay of the state of these blocks being changed by the user so to depict the update effectuated by the dynamic input by the user (e.g. Fig 5; see para 0129-0134, pg. 9-11). It would have been obvious for one skill in the art to provide Simulink5's dynamic display of blocks in the user-driven tool wherein representation of the blocks is listed during a step execution flow as set forth above, so that the effect of the user's input is reflected dynamically on this block execution for this update to be immediately depicted **as a dynamically updated list indicating state of the dynamic update** thereof because this enables the user to assess on the extent of the data being modified, and keep it as preview information needed to impart program code change for future readjusting of any potential prototype with knowledge based on attributes and properties being imparted to the step execution, the output thereof as fed back in form of the dynamic visual update as contemplated in Chandhoke's approach to preview effect of changes and to implement them more efficiently (see para 0137-0167, pg. 10-11, e.g. *updated dynamically, preview ... visually indicate the change* - para 0164, col. 11).

As per claims 2-3, Simulink5 discloses displaying a visual indicator indicating an association between an executing block method and a calling block on the model view (pg. 2-19—2-20, pg. 2-31; pg. 14-24); indicator indicating an association between a currently executing system method and a subsystem block owner (pg. 2-22,23; pg. 2-32; pg. 14-24,25) of said currently executing system method on the model view.

As per claim 4, Simulink5 discloses creating a visual representation of a model component not previously displayed in the model view, the model component calling a method; and displaying a visual indicator indicating an association between the visual representation of the model component not previously displayed and the method called by the model component (*refine output* – pg. 2-17; ch. 10: pg. 10-14→10-20).

As per claim 5, Simulink5 discloses extending a visual indicator from an originating point to a first called method depicted in the model view; and extending sequentially the visual indicator to at least one of each subsequently called method depicted in the model view during a time step in the execution (*propagating*, *link…nonstructural* - pg. 5-28).

As per claims 6-7, Simulink5 discloses indicating a type of method executing in the model view; as a visual indication (ch. 14: pg. 14-24,25).

As per claim 8, Simulink5 discloses visual indication is made by at least one of altering the color of a portion of a model component in the model view representing said method (see pg. 4-5-→4-17; pg. 5-15) and inserting a geometric design (pg. 4-17-4-21, 4-36 – Note: subsystem of sinusoidal functions or entering a diagram representing a circuit for annotating input **reads on** geometric design) in a model component displayed in the model view.

As per claims 9-10, see visible breakpoints in the model view and conditional breakpoints (e.g. ch. 13: *conditional breakpoints* - pg. 13-12 → 13-15; ch. 13: pg. 13-24).

As per claim 11, Simulink5 discloses arranging the execution list view to show the methods executed in a current time step in the execution of the model in a tree structure (tree – ch. 5: pg. 5-18→5-33).

As per claims 12-13, Simulink5 discloses that a user sets visible breakpoints in the execution list view; wherein the breakpoints are conditional breakpoints (see claim 9-10; pg. 13-24).

As per claim 14, Simulink5 discloses setting at least one a trace point and a display point in at least one of the model view and the execution list view (see pg. 13-14-→ 13-26).

As per claims 15-16, Simulink5 discloses generating at least one of debugging data and profiling data (ch. 14: pg. 14-21) during the execution of the model;

associating said at least one of debugging data and profiling data with at least one of said components of the model; and

visually indicating said associated data in the model view (Profile Summary: pg. 14-21→14-27);

wherein said associated data includes solver data (Note: using profile to support data solving with accelerator reads on solver data being associated with model components under profiling execution).

As per claim 17, Simulink5 discloses generating debugging data with said debugger during the execution of the model; associating said debugging data with at least one of said components of the model; and visually indicating said associated data in the execution list view (see ch. 13-17-→13-25).

As per claim 18, Simulink5 discloses the number of iterations of at least one of the plurality of model components during a time step in the execution (e.g. pg. 13-19; pg. 4-42).

As per claims 19-20, Simulink5 discloses selecting a user-set speed parameter via a control associated with the model view; and executing the model in the model view based on the selected speed parameter (pg. 10-41; *parameters dialog box* -pg. 14-6) selecting a user-set speed parameter via a control associated with the execution list view; and executing the model in the execution list view based on the selected speed parameter (Note: setting up accelerator for simulation reads on parameter control associated with execution list – see claim 1).

As per claim 21, Simulink5 discloses receiving input from a user-controlled input device in said graphical modeling and execution environment, said input being interpreted by said graphical modeling and execution environment as a user-selected speed parameter; and executing the model in the execution list view based on the selected speed parameter (refer to claims 19-20 for analogous subject matter based on user input and control parameter at tool graphical level).

As per claims 22-23, Simulink5 discloses altering at least one of a connection between the model components; and adjusting at least one of the execution list view and the model view to indicate the effects of the altering (e.g. ch. 4: pg. 4-9 → 4-19; ch. 2: pg. 2-10→1-16; pg. 6-7→6-24; refer claims 19-20); wherein said altering step includes at least one of adding or removing of at least one of model components and a connection between the model components (ch. 4; pg. 6-7→6-24).

As per claim 24, Simulink5 discloses displaying elements of the compiled state (e.g. *CompiledSampleTime* -pg. 2-28; *compiled model* - pg. 14-4,5) of the model in the model view.

As per claims 25-26, Simulink5 discloses displaying debug information in the model view as a tool tip (e.g. *tooltip* – pg. 3-6) over a component of the model in response to user input; wherein the displayed debug information indicates a signal value (pg. 6-29→6-31) of a signal line in the model view.

As per claims 27-28, Simulink5 discloses wherein the displayed debug information is made persistent in the model view (see pg. 4-76; 5-20; 10-23); wherein said displayed debug information is updated in response to the execution of the model (ch. 2: pg. 2-11→2-19; pg. 4-76).

As per claims 29-31, Simulink5 discloses displaying debug in the execution list view as a tool tip in response to the movement of a pointing device (pg. 3-6; *tooltip* – pg. 14-16; *Navigating, masked* - pg. 9-9, 9-12; *clicking* - pg. 14-26 – Note: tooltip shown as a result of a cursor navigating move during analyzing state of simulation or modeling **reads on** debug information for each block of models being setup or executed as seen in pg. 14-37) in the execution list view over a component of the model associated with said debug information;

wherein the displayed information is made persistent in the execution list view (Note: any data displayed for an instance of simulation is persistent for said list of execution instance); wherein said displayed information is updated in response to the execution of the model (refer to claim 28).

As per claim 32, Simulink5 discloses filtering the displayed execution list of methods in the execution list view so that only methods satisfying (ch. 9: pg. 9-2→9-7) a user-specified criteria are displayed.

As per claims 33-34, see (pg. 4-70, 79; pg. 14-25) for creating a record for a unique method invocation; and displaying data associated with the unique method invocations as the unique invocation is called; anchoring said record to a block owner of (*clicking*- pg. 14-24→14-26; pg. 9-12; pg. 13-21,23) said unique method invocation in the model view (Note: one parent block reads on method invocation being unique).

As per claims 35-36, Simulink5 discloses displaying the calling of said unique method invocation with varying degrees of intensity representative of the frequency of the invocation (ch. 14: pg. 14-24→14-25); creating a unique method invocation for an execution exception event (*error message, error dialogue* -- ch. 2: pg. 2-23, 24; pg. 7-13,14; pg. 10-36).

As per claim 37, Simulink5 discloses wherein a user sets non-visible breakpoints (ch. 13: pg. 13-24 – Note: programmatic breakpoints being conditional to execution reads on non-visible) in at least one of the model view or the execution list view.

As per claim 38, Simulink5 discloses wherein at least one of a set of debugging data or a set of profiling data are displayed to a user in a separate view (*help browser* – pg. 14-23).

As per claim 39, Simulink5 discloses a medium holding computer-executable instructions for performing debugging in a graphical modeling and execution environment on an electronic device, said medium executable on said electronic device for performing a method, said method comprising instructions:

providing a model view and an execution list view an execution list depicting the execution order of methods called … time step … with a debugger;

indicating visually the state of the execution … model view;

all of which steps having been addressed in claim 1 which recites the same corresponding limitations.

Simulink5 does not explicitly teach that graphical view of the execution list depicted during the execution of a time step is such that the execution list is dynamically updated and changing during the execution of the model to list the methods that have been called, nor does Simulink5 disclose indicating a state of said dynamically updated execution list. But this limitation has been addressed in claim 1 as obvious from above.

As per claims 40, 41 and 42-43, refer to rejection of claims 2, 3, and 5, respectively.

As per claims 44-76, refer to claims 6-38 respectively for corresponding rejection.

As per claim 77, Simulink5 discloses in a graphical design environment, a system comprising:

a debugger storage, said debugger gathering debug information from the simulation of a model in said graphical design environment (ch. 2; ch. 13); a display device in communication with the design environment, said device displaying:

a model view, the model view displaying a plurality of components of a model and being interfaced with said debugger; and

an execution list view, the execution list view displaying an execution list (ch. 2: pg. 2-10→2-24; ch.13: pg. 13-17→13-19) depicting an execution order of methods called during the execution of a time step of the model, the execution list view state being visually represented (*Time Step, Math Function block, Sum block, Product block* - pg. 2-10,11; 2-19,20; ch. 5: pg. 5-16→5-17) on the model view, the execution list view being generated by said debugger.

However, Simulink5 does not explicitly teach that graphical view of the execution list depicted during the execution of a time step is such that the execution list is dynamically updated and changing during the execution of the model to list the methods that have been called, nor does Simulink5 disclose indicating a state of said dynamically updated execution list. But this limitation has been addressed in claim 1 as obvious in combination with Chandhoke from above.

As per claims 78-79, refer to claims 2-3 (refer to claim 1 for block order).

As per claim 80, refer to claims 12 and 14;

As per claims 81-83, refer to claims 13, 6, and 8 respectively for corresponding rejection.

Response to Arguments

7. Applicant's arguments filed 1/09/08 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 103(a) Rejection:

(A) Applicants have submitted that the cited portions in Simulink5 discuss controlling execution order of blocks and display of block execution, not display as claimed of list of methods called, let alone execution order of method called during a time step (Appl. Rmrks pg. 15, middle). The listing of method calls and the 'execution order' display has been interpreted in light of the Specifications, and when the claim (e.g. claims 1, 39) language referring to the 'dynamically updated execution list' specifies it as 'execution order' being depicted AND 'list of method calls', both being in conjunction with step execution, portions of the Disclosure related to step execution, *list of method calls*, and *execution order* have been analyzed to understand the dynamic upgrade aspect of said list or execution order in light of said model step execution in a specific timeframe. As set forth in the Claim Objection, the language describing 'dynamically updated execution list' for both 'execution order' and 'list of method calls' that have been called does not have reasonable support from the Disclosure. The Disclosure does not reasonably convey that time-line execution of method blocks as displayed (see Specs: Fig. 5, 7-9 and Claim Objections) wherein blocks 'execution order' can be viewed and reconfigured, is exactly the same 'directory view' (see Specs: Fig. 17-18) of highlighted areas of a tree representing method calls, which can be manipulated by the user dynamically along with the debugger to continually upgrade a calls record, i.e. nothing related to time step invoking of methods and display of *execution order* therefor. As set forth in the Objections to the claim language, the disclosure rather depicts execution order in terms of visually displayed block representing methods, and distinct from that, colored areas representing method call nodes among a directory tree are visual display of evolving methods which can be manipulated by the user. Hence, 'dynamically updated execution list' cannot be interpreted as being both 'execution order' and 'list of method calls' in the context of model execution via some interval time stepping. Because of the improper use of the language (see Claim Objections), the *method calls* that have been called during execution of a model step instance have to bear or take the interpreted connotation of (a) method invocations underlying the iconic blocks being viewed by the developer as the execution is in progress, and (b) all the user invocations or dynamic directive/reconfiguring that enable the model view to step execute and show results from the method block being executed as displayed on the simulation interface. The rejection has pointed out where method blocks execution is visually displayed during runtime, and where a listing of user invoked methods support dynamic yielding of change of state regarding the elements composing the model being simulated. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference. The argument is therefore not persuasive.

(B) Applicants have submitted that Chandhoke nowhere disclose ‘dynamically updated execution list … execution order … a time step of a model’ as presented in claim 1, nor does Chandhoke disclose 'dynamically updated execution list' to list methods that have been called during the time step (Appl. Rmrks, pg. 15 bottom, pg. 16 top). The above section has analyzed how the rejected has interpreted the ‘dynamically updated execution list' and based thereupon (see Claim Objections) has cited portions in Simulink5 to match with the 'execution order' and the list of method calls. That is, *execution order* depiction is been analogized to **method** blocks being visually shown (Simulink5, chp. 5) with underlying computation execution to effectuate input outputs, the like of which computation settings being subject to user's intervention to change to result of the methods underlying those method blocks. That is, methods called during a time step would become all the directives by the user to reconfigure the outcome of the model execution, and such list of directive or code commands have been cited using chapter 13 of Simulink5. The rationale as to why Chandhoke is applicable in the USC 103 rationale is not remotely aimed at addressing or fulfilling the above 'dynamically updated execution list' limitation as raised above, and the belief the Chandhoke does not remedy to Simulink5 regarding that limitation, seems to be largely misplaced. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(C) Applicants have submitted that for claim 39, Simulink5 in view of Chandhoke taken in combination cannot disclose or suggest 'execution list depicting a dynamically updated … of a time step of a model … updated execution list ... to list methods that have been called ... point in execution of the time step' (Appl. Rmrks, pg. 16 to 17). The crux of the argument falls under the ambit of the analysis set forth in the above sections, incorporating the analysis set forth in the Claim Objection regarding inaccurate use of language; hence is referred back to sections A, B from above.

(D) As for the argument regarding the same limitation being repeated in claim 77, refer to the sections A, B including the analysis set forth in the Claims Objections.

The claims will stand rejected as set forth in the Office Action.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

April 02, 2008